# 一种基于支持向量机的跨站脚本漏洞检测技术 *

黄娜娜 a, b，万　良 a, b†

(贵州大学 a. 计算机科学与技术学院; b. 计算机科学理论研究所，贵阳 550025)

**摘　要：**跨站脚本是一种常见的针对 Web 应用程序安全的漏洞攻击方式。恶意用户利用漏洞将恶意脚本注入网页之中，当用户浏览该网页时，便会触发脚本，导致攻击行为产生。为此，针对各种变形跨站脚本攻击难以检测问题，对一种基于正则表达式和支持向量机的递归特征消去算法（RE-SVM-RFE）进行了研究。首先采用正则表达式匹配算法，为训练集选择有代表性的特征，即对数据预处理；再利用 RE-SVM-RFE 特征选择算法选择出最优特征，再对具有攻击性的关键词进行特征排序；最后通过总结特征关键字的出现频率，发现频率越高漏洞存在可能性越大。实验结果表明，数据经过 RE-SVM-RFE 递归特征消去算法选择之后的 SVM 特征，预测的准确率更高，敏感度和特异度也更好，该算法能够有效地检测出跨站脚本漏洞。

**关键词：**支持向量机；跨站脚本攻击；特征向量；Web 安全；特征选择；RE-SVM-RFE 算法

**中图分类号：**TP309.02　　doi: 10.3969/j.issn.1001-3695.2017.08.0712

## Cross site scripting vulnerabilities detection based on support vector machine (SVM) technology

Huang Nana[a, b], Wang Liang[a, b†]

(*a. College of Computer Science & Technology*, *b. Institute of Computer Science Guizhou University*, *Guiyang* 550025, *China*)

**Abstract:** Cross-site scripting is a common way of exploiting Web application security vulnerabilities. A malicious user exploits a vulnerability to inject a malicious script into a web page, and when the user browses the page, it triggers the script, causing the attack to occur. This paper studied a recursive feature elimination algorithm based on regular expression and support vector machine (RE-SVM-RFE) for each kind of deformation. Firstly, the regular expression matching algorithm, to select a representative training set of characteristics, i. e. , the data preprocessing; reuse RE-SVM-RFE feature selection algorithm to select the optimal characteristics, and then the keyword feature offensive sort. Finally, it summarized the frequency of occurrence of the keyword feature, found that the higher the frequency the greater the likelihood loopholes. Experimental results show that, after the data RE-SVM-RFE recursive feature elimination after SVM feature selection algorithm, higher accuracy of prediction, and better sensitivity and specificity, the algorithm can effectively detect XSS.

**Key Words:** SVM; XSS; feature vectors; Web security; feature selection; RE-SVM-REF algorithm

## 0　引言

Web 商业化以来，Web 发展一直在不断增长，Web 以其自身的开放性和易用性受到越来越多的开发者和用户关注。然而，正在 Web 为人类生活提供便捷时，其应用程序中所包含的各类漏洞[1]也随之成为 Web 上最为严重的安全隐患之一。跨站脚本(cross site scripting，XSS)[2]是数据通过不受信任的来源（通常是 Web 请求）进入 Web 应用程序，数据被包含在动态内容中，发送到 Web 用户，而恶意脚本内容不会被验证。当最终用户浏览网页时，可能会触发未被验证的恶意脚本代码，从而攻击者会发送私有数据（如 Cookie 或其他会话信息），将受害者重定向到由攻击者控制的 Web 内容。

关于跨站脚本的检测最早可追溯到上个世纪，随后关于该研究的研究者们提出很多方法。最早对跨站脚本进行测试的方法是渗透测试[3]，在渗透测试进行的过程中，把渗透测试的脆弱性分析和动态分析相结合，有效利用扩展的污染模式模型来检测 XSS 的存在情况。文献[4]提出了基于行为对比方法来进行漏洞的判定，通过系统在普通状态下和受攻击状态下的行为对比来判断系统中的漏洞存在情况。文献[5]设计一个保护 Web 服务器的入侵检测工具，能够实时跟踪可疑主机，能检测出恶意的 XSS，但是其通用性不强，模式匹配部署工具中只包含对 Apache Web 这一服务器的检测。文献[6]提出一种使用软件故

---

**基金项目：**贵州省科学基金资助项目（黔科合 J 字［2011］2328 号；黔科合 LH 字［2014］7634 号）

**作者简介：**黄娜娜（1986-），女，江苏宿迁人，硕士研究生，主要研究方向为 Web 应用安全漏洞、信息安全；万良（1974-），男（土家族，通信作者），贵州铜仁人，教授，博士，主要研究方向为形式化方法、信息安全（wanliangtr@163.com）.9/10.

障注入技术来检测自动 Web 漏洞扫描程序的方法，此方法通过最常见的软件故障类型注入到 Web 应用程序代码中，然后由扫描程序进行检查，检查出存在跨站脚本攻击漏洞，但是此方法的漏洞检测覆盖率低，误报率较高。文献[7]提出一种 2 阶段的静态检测方法来寻找并移除服务器端代码中的 XSS 漏洞，该方法在第 1 阶段采用污点分析方法来跟踪用户输入，确定潜在的脆弱点，第 2 阶段采用模式匹配和数据依赖性分析找出源码中存在漏洞的位置并对漏洞进行移除，该方法对服务端代码分析检测效果较好，但由于没有实现对客户端代码的检测，因此无法检测出 DOM 的 XSS。文献[8]提出一种使用最优攻击向量的 XSS 漏洞检测方法来检测 Web 应用程序中的 XSS 漏洞，此方法自动生成矢量图，优化模型的攻击向量计数器，但是该方法需要进行长时间的学习。

已经有许多方法用来检测 XSS 攻击了，但是由于都有各自的缺点，目前很少有研究者借助支持向量机的分类器[9,10]来检测 XSS 漏洞。因此，本文在借助支持向量机分类器检测的基础上，研究出一种把正则表达式匹配算法和支持向量机的特征消除选择算法进行结合的特征重组算法(RegEx and recursive feature elimination based on support vector machine, RE-SVM-RFE)，针对跨站脚本漏洞进行检测。实验首先要收集正常页面和存在 XSS 漏洞的页面 Web 请求数据集，同时分别对两类页面中的 URL、JavaScript 代码以及 Post 请求等进行特征提取建立样本数据集，通过上面步骤中收集的载荷数据，然后利用正则表达式匹配算法对原始样本数据集进行预处理，提取出最优的特征，随后进行特征的组合筛选，形成特征数据集，利用 SVM 算法的模型训练传递到分类器中进行结果检测，最终给出检测结果。XSS 攻击检测流程如图 1 所示。

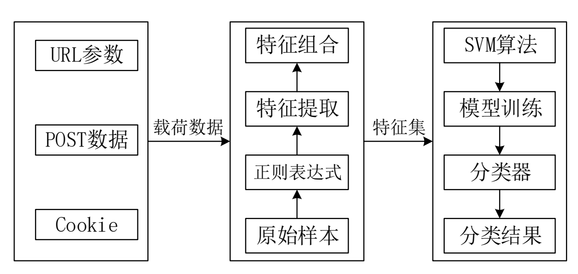



图 1　XSS 攻击检测流程

## 1　支持向量机基本原理

SVM 的原理是寻找一个满足分类要求的最优分类超平面[11]，使得该超平面在保证分类精度的同时，能够使超平面两侧的空白区域最大化。理论上，支持向量机能够实现对线性可分数据的最优分类。SVM 是从线性可分情况下的最佳分类超平面发展而来，其基本思想可用如图 2 所示的情况来解释。现假设给定两类数据分类的训练样本集为 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$，$y_i\in\{-1,+1\}$，其中 $x_i$ 是原始样本的特征向量，$y_i$ 是相关联的类标号；在二分类中，每个 $y_i$（即 $y_i\in\{-1,+1\}$）取二值之一，表示是否属于这个类。如果有线性函数能将两类样本完全分开，就称为线性可分，否则为非线性可分。最佳超平面如图 2 所示。

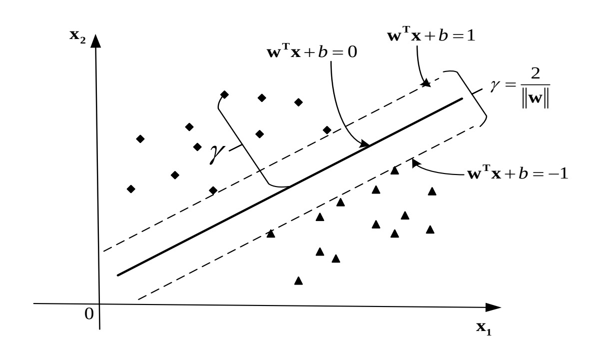



图 2　支持向量机最佳超平面

## 2　RE-SVM-RFE 特征选择算法

特征优选的目的是将最有价值的、彼此相关性不强的检测特征保留下来。对于两类分类问题，意味着两类样本之间的平均相似性更小，具有更大的可分性。因此，可以定义类别可分性度量用于表示两类样本的差异。类别可分性越大，则两类样本间的相似性越小，越容易分开。特征选择和提取就是从候选特征集中选出或提取与任务最相关的特征集。特征选择的基本框架如图 3 所示。

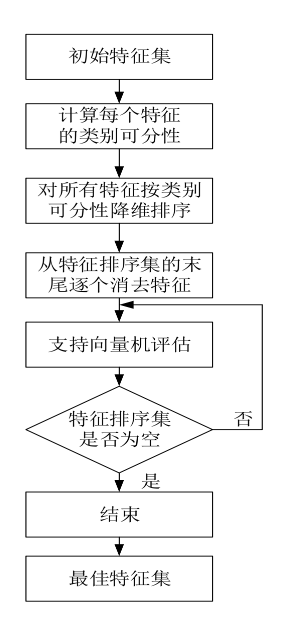



图 3　特征选择的基本框架

### 2.1　正则表达式匹配算法

正则表达式以其强大和灵活的表达能力，已成为描述新一代规则的主要工具。使用正则表达式来描述攻击的特征，比传统的提取精确字符串方法更准确、方便和有效。XSS 攻击与程序中的字符串变量的操作有关，通过注入脚本代码带来安全隐患。本文为了能够做到分析数据集中字符串变量的取值集合，字符串变量在数据集中经过一系列的定义、赋值以及组合操作，使用正则表达式匹配算法来表示其在不同数据集中的取值，这使得正则表达式匹配算法成为属性选择的最好算法。

## 2.2 RE-SVM-RFE 算法

RE-SVM-RFE 算法的提出是根据支持向量机递归特征消除算法[12]的思想，在特征筛选过程中进行了一些改进。在 RE-SVM-RFE 算法处理数据集前，首先利用正则表达式匹配算法从数据集中挑选出有代表性的特征数据，本实验从数据集中挑选出了 46 个有代表性的特征；然后通过迭代训练选出一些最优特征。对选出的良好特征进行分组测试，利用递归特征消除算法计算出组合特征权重[13]值，通过每次迭代都能对组合特征的权重值进行排序，筛选出权重值最小的进行删除。在开始步骤中特征集合是比较庞大的，但是随着算法的进行，每次的迭代都会移除一个组合特征，本实验通过 RE-SVM-RFE 算法迭代后，最终挑选出了 6 个最优的特征。对挑选出的 6 个最优特征，进行重新组合来测试 RE-SVM-RFE 算法与 SVM-RFE 算法的差别。

### 2.2.1 RE-SVM-RFE 算法的迭代过程

在 RE-SVM-RFE 算法的执行过程中，每次都需要挑选出一个权重最小值，然后将其移除。在算法每删除一个权重最小值时都有三个执行步骤，具体内容如下：

a)用样本数据集训练分类器，得到的特征数据要与分类器的特征相关（如优化权值 $\omega_h$ ）；

b)依据前面正则表达式匹配算法挑选出的特征来进行筛选同时计算挑选出的 46 特征的属性值（如代价函数 $DJ(h)$ ）；

c)删除数据集中挑选出的最小权重，即最小排序标准的特征。

在最优特征的选择过程中，每选择出一个最优特征，都是要进行一定的迭代过程，直到数据集中只留下一个特征变量时方可结束，最终选择出的结果是获得了一列按照特征重要性排列的特征排序列表，本实验中，最终选出 6 个最优特征。下面介绍 SVM 的代价函数。

对于线性 SVM，消去第 $h$ 个特征的代价函数：

$$DJ(h) = (\omega_h)^2 \tag{1}$$

对于非线性 SVM：

$$DJ(h) = \frac{1}{2}\alpha^T H\alpha - \frac{1}{2}\alpha^T H(-h)\alpha \tag{2}$$

$H$ 是一个矩阵，其元素为 $y_i y_j K(X_i, X_j)$ ，$H(-h)$ 为消去第 $h$ 个特征后的矩阵。其中 $K$ 表示核函数，核函数测量实例 $X_i$ 与 $X_j$ 之间的相似性。

### 2.2.2 RE-SVM-RFE 算法过程

RE-SVM-RFE 算法的整体过程如下所示：

输入：

训练样本矩阵： $X_0 = [x_1, x_2, \cdots, x_i, \cdots x_n]^T$

类别标签： $y = [y_1, y_2, \cdots, y_i, \cdots, y_n]^T$

初始化：

当前特征子集向量： $s = [1, 2, \cdots, k]$

特征排序向量： $r = [\ ]$

特征排序：

迭代过程直到： $s = [\ ]$

获取当前新的训练样本矩阵： $X = X_0(:, s)$

给定参数后训练分类器： $\alpha = SVM - \mathrm{train}(X, y)$

计算权值向量： $w = \sum_i \alpha_i y_i x_i$

计算排序标准： $c_h = DJ(h)$

寻找特征排序得分值最小项： $f = \arg\ \min(c)$

更新特征排序向量： $r = [s(f), r]$

消去得分最小特征： $s = s(1: f-1, f+1: length(s))$

输出：特征排序列表 $r$ 。

经过上面 RE-SVM-RFE 算法的筛选过程最终得到最优的特征排序表。排序在前面的单个特征不一定能使得 SVM 分类器[14]有很好的分类性能，需要将多个特征进行组合，才能使得 SVM 分类器有最优的分类性能，要想得到较好的分类结果需先对 SVM 模型进行训练。首先利用挑选出来的特征排序表定义一定数量的嵌套的特征子集 $F_1 \subset F_2 \subset \cdots \subset F_n$ 来训练 SVM 模型[15]，然后用 SVM 模型来预测正确率来准确评估这些特征子集的优劣，最终能够获得最优的特征子集。挑选结果说明，经过 RE-SVM-RFE 算法选择后，能挑选出最优的特征。

在用 RE-SVM-RFE 算法选择最优特征子集时，本文用 10 折交叉验证算法[16]，用固定的参数维数来分配训练集和测试集。在特征排序表训练过程中使用 SVM 参数寻优方法进行参数寻优。具体详细的特征选择框架如所图 4 所示。

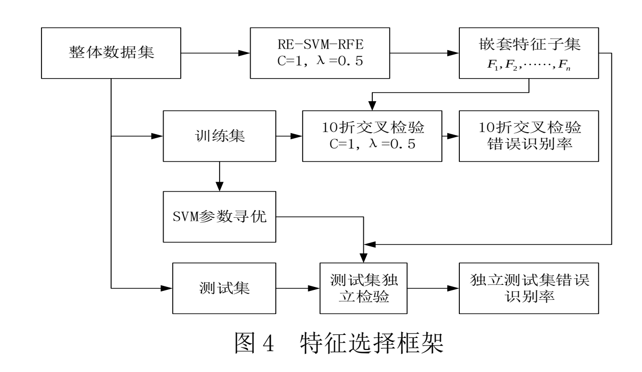



图 4 特征选择框架

## 2.3 RE-SVM-RFE 算法的数据处理结果

从超文本传输协议请求数据包中提取的载荷数据都是非结构化的，需要进行结构化处理，从原始特征中挑选出数量为 $n$ 的一组最优特征，在不降低分类准确率的前提下降低原始特征的空间维数，将原始载荷数据转换为固定维数的特征向量，作为 XSS 攻击检测算法的输入数据。

通过分析漏洞验证阶段、漏洞利用阶段及各种绕过情景下的 XSS 攻击语句的各种形式，采用人工挑选、数学统计以及 RE-SVM-RFE 算法相结合的方式对原始载荷数据进行特征选择，经过不断的迭代，再从选择的特征中进行分组测试，最终选择出最优的 6 个特征：各种截断、闭合等特殊字符频率（特殊字符个数/字符数）、大写字母字符频率；攻击特征关键字频率、小写字母字符频率、数字字符个数频率和空格字符频率。特征的汇总和具体含义如表 1 所示。

表 1　特征名称与含义

| 特征名称 | 特征含义 |
|---|---|
| 攻击特征关键字 | 相应类型攻击的特征关键字 |
| 特殊字符频率 | 各种闭合、截断等特殊字符 |
| 大字字母字符频率 | 大字字母(A-Z) |
| 小写字母字符频率 | 小字字母(a-z) |
| 数字字符频率 | 数字 0-9 |
| 空格字符频率 | 空格字符频率 |

其中，XSS 攻击语句中常见的攻击特征关键字有 script、java、iframe、alert、img、style、prompt、location、hash、src、href、eval、XMLHttpRequest、ActiveXobject、@improt；常见的特殊字符有：<、>、"、'、%、(、)、!、#、&、:、=、?、@、[、]、/、\、{、}、|、$、,、*、+、-、;、.。

由于实验数据包含两类样本，为了对数据集进一步细致的分析和理解，筛选出具有显著区分能力的潜在特征。即从 46 个特征中挑选出 6 个最优特征进行分类组合，再用 SVM-RFE 和 RE-SVM-RFE 的 10 折 10 倍交叉验证方法的进行平均分类准确率和标准偏差计算。结果对比如表 2 所示。

表 2　两种算法的准确率对比

| | SVM-RFE /% | RE-SVM-RFE /% |
|---|---|---|
| 特征 1 vs 特征 2 | 77.96 ± 3.29 | 83.69 ± 3.07 |
| 特征 2 vs 特征 3 | 78.82 ± 3.67 | 78.82 ± 3.19 |
| 特征 3 vs 特征 4 | 78.82 ± 4.48 | 86.18 ± 2.78 |
| 特征 3 vs 特征 4 vs 特征 5 | 72.00 ± 4.15 | 78.56 ± 2.88 |

从表2中可以看出，无论从平均准确率还是标准偏差上RE-SVM-RFE 的结果均好于 SVM-RFE。在特征 1 vs 特征 2、特征 3 vs 特征 4 和特征 3 vs 特征 4 vs 特征 5 三组分类问题上，RE-SVM-RFE 的分类准确率比 SVM-RFE 分别高 5.73%、7.36%和 6.56%。在特征 2 vs 特征 3 分类问题上，虽然两者的分类准确率相同，但 RE-SVM-RFE 的标准偏差明显小于 SVM-RFE，即 RE-SVM-RFE 算法得到的结果更稳定，如表 3 所示。

表 3　敏感度和特异度的比较

| | SVM-RFE Sensibility(%) | RE-SVM-RFE Sensibility(%) | SVM-RFE Specificity(%) | RE-SVM-RFE Specificity(%) |
|---|---|---|---|---|
| 1 组 | 75.90 ± 4.63 | 81.53 ± 5.83 | 76.97 ± 5.02 | 85.00 ± 3.92 |
| 2 组 | 78.09 ± 3.29 | 79.05 ± 3.01 | 79.87 ± 5.94 | 82.87 ± 4.03 |
| 3 组 | 76.87 ± 5.98 | 84.25 ± 5.46 | 80.23 ± 5.63 | 88.63 ± 5.09 |
| 4 组 | 76.96 ± 3.09 | 73.00 ± 3.85 | 76.06 ± 3.19 | 79.56 ± 2.68 |

同时，为了弥补只考虑准确率的不足，对两类问题，敏感度（sensitivity）和特异度（specificity）也是一种常用的度量指标。表 3 中的 1 组、2 组、3 组和 4 组分别代表特征 1 vs 特征 2、特征 2 vs 特征 3、特征 3 vs 特征 4 和特征 4 vs 特征 5 这 4 组。表 3 中给出了两种算法在敏感度和差异度上的比较。可以看出，RE-SVM-RFE 算法的结果要明显好于 SVM-RFE 的结果。

算法执行结果说明，RE-SVM-RFE 算法选出了区分能力更为显著的特征，即选出了对类别区分能力很强的特征，XSS 攻击样本数据中攻击特征关键字确实能够影响 SVM 模型的建立，从而对特征评价产生影响，因此，在 SVM-RFE 特征选择之前通过正则表达式进行数据预处理是非常有意义的。

## 3　实验与结果分析

### 3.1　实验准备

本文实验使用数据集均来自互联网，但是有两个不同的方面。正常样本数据集是通过分析 Web 服务器日志提取良性访问资源的请求得到的，跨站脚本攻击样本数据集是通过分析漏洞提交网站 XSSED[17]、HA.CKKERS[18]与 exploit-db.[19]得到的。本实验所用数据集共有两部分组成：正常的载荷请求样本集和跨站脚本漏洞样本集。在之前的准备工作中，共收集 1 378 条样本数据，样本集如表 4 所示。

表 4　样本集分类

| 请求类型 | 样本数量 |
|---|---|
| XSS 请求 | 686 |
| HTTP 正常请求 | 692 |

本实验用到的主机配置为为 CPU Intel I5，主频 2.0 GB，内存 8 GB，操作系统是 Win10 64 位。对于现有的机器识别方法，采用 Weka 实验平台进行测试，通过 SVM 分类器来识别跨站脚本攻击。本文提出的算法则是通过 Microsoft Visual Studio 2015 与 Microsoft SQL Server 2012 编程进行数据集的特征预处理，首先对收集到的原始数据进行分类概括总结出了 46 个特征，然后利用正则表达式匹配算法进行数据的预处理，再采用本文提出的 RE-SVM-RFE 特征选择算法来匹配出最好的特征。从 46 个特征中找出最具代表性的 6 个特征作为最终的验证数据集。最佳特征应具有稳定性、可辨别性和相对独立性等特征，其中稳定性表示同一类别的特征值就相近、可辨别性表示不同类别的特征取值应具有明显差异、相对独立性表示各个不同特征之间关联性不强[20]。本实验中提取出的 6 个最优特征能够准确的反映 Web 请求载荷数据的本质特征。表 5、6 给出了提取后的最优样本特征值。每个特征所表示的含义为：特征 1 表示特殊字符频率、特征 2 表示数字个数频率、特征 3 表示小写字母个数频率、特征 4 表示大写字母字符个数频率、特征 5 表示不同类型的攻击关键词特征字符频率和特征 6 表示载荷数据中是否出现相应类型攻击的特征关键字类别标号。最后选择出最优的各类样本集特征如表 5、6 所示。

从表 5、6 所列的正常样本和跨站脚本攻击样本中，可以看出同一类别的特征值取值相近，不同特征的特征值取值相差较大，这说明提取的数据特征具有强的可辨别性、高的稳定性，然而不同的特征之间没有相关性，说明数据特征具有很好的独立性。

表 5　正常样本特征

| 特征 1 | 特征 2 | 特征 3 | 特征 4 | 特征 5 | 特征 6 |
|---|---|---|---|---|---|
| 0.1389 | 0.2083 | 0.5694 | 0.0556 | 0.0000 | 0 |
| 0.2400 | 0.0000 | 0.7600 | 0.0000 | 0.0000 | 0 |
| 0.2857 | 0.0000 | 0.7143 | 0.0000 | 0.0000 | 0 |
| 0.3182 | 0.0000 | 0.6818 | 0.0000 | 0.0000 | 0 |
| 0.1719 | 0.4219 | 0.3906 | 0.0000 | 0.0000 | 0 |

表 6　XSS 攻击样本特征

| 特征 1 | 特征 2 | 特征 3 | 特征 4 | 特征 5 | 特征 6 |
|---|---|---|---|---|---|
| 0.2045 | 0.0114 | 0.7614 | 0.0000 | 0.0341 | 1 |
| 0.2119 | 0.0593 | 0.6695 | 0.0508 | 0.0254 | 1 |
| 0.1686 | 0.1065 | 0.6391 | 0.0680 | 0.0118 | 1 |
| 0.1684 | 0.1053 | 0.6737 | 0.0421 | 0.0316 | 1 |
| 0.2000 | 0.1517 | 0.6276 | 0.0207 | 0.0138 | 1 |

### 3.2　结果分析

为了验证该算法的正确性，需要用到一些评估指标，如混淆矩阵、查准率、查全率、ROC 曲线和 F-measure 值。混淆矩阵中所有参数及参数意义为：真正例（true positive，TP），表示实际为正类被预测为正类样本的个数；假正例（false positive，FP），表示实际为负类被测为正类的样本个数；真反例（true negative，TN），表示实际为负类被预测为负类的样本个数；假反例（false negative，FN），表示实际为正类被预测为负类的样例个数，最终样本总数=TP+FP+TN+FN。分类结果的混淆矩阵如表 7 所示。

表 7　分类结果混淆矩阵

| 真实情况 | 预测结果 | |
|---|---|---|
| | 正例 | 反例 |
| 正例 | TP（真正例） | FN（假反例） |
| 反例 | FP（假正例） | TN（真反例） |

本文提出的算法最终得到的混淆矩阵数据如表 8 所示。

表 8　本文算法分类结果混淆矩阵

| 真实情况 | 预测结果 | |
|---|---|---|
| | 正例 | 反例 |
| 正例 | 632 | 64 |
| 反例 | 32 | 660 |

从表 8 中可以看到主对角线的检测出来的数据远远大于副对角线上的数据，这说明本文所用方法得到的准确率较高。

查准率（precision）和查全率（recall）一般是不会被孤立的进行讨论的，但是有时候为了检测的方便也会单独测试，本实验验证算法则是单独测试的，查准率 P 和查全率 R 分别表示如下：

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$P = \frac{TP}{TP + FP} \tag{4}$$

在检测跨站脚本攻击实验中，完美的查准率值应为 1.0 意味着通过检测的每个结果都是相关的，完美的查全率值也为 1.0 意味着所有存在漏洞的请求数据都被检测出来，本文所提出算法的测试结果如表 9 所示，检测结果明显优于其他算法。

ROC 曲线[21,22]，是用来衡量学习训练器泛化性能的工具，ROC 曲线是由学习训练器计算出的两个重要的特征量决定。ROC 曲线是用坐标轴来表示的，它的横坐标表示假正例率，它的纵坐标表示真正例率，两者表示如下：

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

$$FPR = \frac{FP}{TN + FP} \tag{6}$$

ROC 曲线描述的是真正例率和假正例率的综合指标。ROC 曲线下与坐标轴围成的面积值（ROC area）越靠近 1，表明对该类数据的检测准确率越高，表 9 中 ROC area 值表现最好的则是本文算法，说明本文算法对跨站脚本攻击有较高的检测率。

在前人的研究中，一般采用设定固定阈值的方法进行特征选择，但是由于阈值的设置缺乏客观性很难实现特征子集的最优选取[23]。本文利用递归反向特征剔除算法每次从特征排序集 R 中删除一个当前 CS 值最小的特征，再利用 SVM 算法对选取出来的特征子集进行评估选值。每次迭代中，采用 SVM 的 F-measure 值作为当前被选特征子集的评估标准。F-measure 是用来度量测试精度，通常被定义为准确率和查全率的加权调和平均值。F-measure 的计算公式 [24] 如下：

$$F - measure = \frac{2 \times \mathrm{Precison} \times \mathrm{Recall}}{\mathrm{Precison} + \mathrm{Recall}} \tag{7}$$

如表 9 所示，F-measure 值表现最好的则支持向量机分类器，说明本文算法对 XSS 攻击有较高的识别率。

表 9　各分类器检测结果数据

| Classifier | Class | TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area |
|---|---|---|---|---|---|---|---|
| REB Tree | +1 | 0.905 | 0.058 | 0.939 | 0.905 | 0.922 | 0.955 |
| | -1 | 0.942 | 0.095 | 0.909 | 0.942 | 0.925 | 0.955 |
| Naive Bayes | +1 | 0.792 | 0.066 | 0.922 | 0.792 | 0.852 | 0.916 |
| | -1 | 0.934 | 0.208 | 0.819 | 0.934 | 0.872 | 0.916 |
| J48 | +1 | 0.794 | 0.081 | 0.907 | 0.794 | 0.847 | 0.857 |
| | -1 | 0.919 | 0.206 | 0.819 | 0.919 | 0.866 | 0.857 |
| Logistic | +1 | 0.847 | 0.053 | 0.940 | 0.847 | 0.891 | 0.945 |
| | -1 | 0.947 | 0.153 | 0.862 | 0.947 | 0.902 | 0.962 |
| LogitBoost | +1 | 0.847 | 0.053 | 0.940 | 0.847 | 0.891 | 0.962 |
| | -1 | 0.947 | 0.153 | 0.862 | 0.947 | 0.902 | 0.962 |
| 本文方法 | +1 | 0.907 | 0.046 | 0.951 | 0.907 | 0.928 | 0.969 |
| | -1 | 0.954 | 0.093 | 0.912 | 0.954 | 0.932 | 0.956 |

不同的分类器消耗时间如表 10 所示。

表 10 各分类器耗费时间的具体数据

| Classifier | Train Time(S) | Test Time(S) | Total Time(S) |
|---|---|---|---|
| REBTree | 0.05 | 0.01 | 0.06 |
| NaiveBayes | 0.03 | 0.02 | 0.05 |
| J48 | 0.07 | 0.04 | 0.11 |
| Logistic | 0.07 | 0.05 | 0.12 |
| LogitBoost | 0.06 | 0.04 | 0.10 |
| 本文方法 | 0.03 | 0.01 | 0.04 |

表 10 中 J48 分类器训练与测试总耗费时间为 0.11 s，但其准确率为 85.709 3%，NaiveBayes 分类器训练与测试总耗费时间为 0.05 s，准确率为 86.284 5%，本文算法训练所用时间为 0.04 s，仅次于 NaiveBayes 的 0.01 s，但识别准确率比 NaiveBayes 高 6.794 4%。训练所耗时间还取决于算法优化，SVM 算法优化程度较好，运行所耗费时间短，RE-SVM-RFE 算法的时间复杂度可以再优化。

## 4 结束语

本文的前期工作量比较大，主要工作要分析跨站脚本漏洞的一些主要特征，了解漏洞产生的原因；然后分析正常用户的输入与攻击者输入的攻击语句之间的不同，要能正确区分出正常请求与跨站脚本攻击请求后；最后收集整理数据集对跨站脚本攻击进行检测。本文基于支持向量机模型训练的原理，提出了一种基于正则表达式匹配算法和支持向量机的特征消除选择算法，通过正则表达式匹配算法对数据集进行预处理，用核函数对预处理后的数据进一步优化，再依据序列最小优化分类器来对跨站脚本漏洞进行检测，最后利用 SVM 模型进行验证检测结果。实验结果表明，本文提出的特征选择消除算法进行分析对比得出对未知的跨站脚本攻击有较好的检测效率。接下来的任务主要还是对数据集的特征进行再优化。本文研究内容主要是建立在实验的基础上，在实际检测应用中还需加以改进和不断的完善。

## 参考文献：


[1] Garg A, Singh S. A review on Web application security vulnerabilities [J]. Internation Journal, 2013, 3 (1): 222-226.

[2] Antunes N, Vieira M. Defending against Web application vnlnerabilities [J]. Computer, 2012, 45 (2): 66-72.

[3] Melbourne J, Jorm D. Penetration testing for Web applications [C]// Proc of IEEE Symposium on Security and Privacy. 2014: 256-263.

[4] Huang Yaowen, Huang Shihkun, Lin T S P, et al. Web application security assessment by fault injection and behavior monitoring [C]// Proc of the 12th International Conference on World Wide Web. 2003: 148-159.

[5] Almgren M, Debar H, Dacier M, et al. A lightweight tool for detecting Web server attacks [C]// Proc of Netword and Distributed Systems Security. 2000: 157-170.

[6] Williams J, Wichers D. OWASP top 10, the ten most critical Web application security risks [R]. New York: The Open Web Application Security Project, 2013.

[7] Shar L K, Tan H B K. Auto-mated removal of cross site scripting vulnerabilities in Web application [J]. Infromation and Software Technolohy, 2012, 54 (5): 467-478.

[8] Guo X, Jin S, Zhang Y. XSS vulnerability detection using optimized attack vector repertory [C]// Proc of IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. 2015: 29-36.

[9] Vishnu B A, Jevitha K P. Prediction of cross-site scripting attack using machine learning algorithms [C]// Proc of International Conference on Interdisciplinary Advances in Applied Computing. New York: ACM Press, 2014.

[10] Keerthi S S, Shevade S K, Bhattacharyya C, et al. Improvements to platt's SMO algorithm for SVM classifier design [J]. Neural Computation, 2001, 13 (3): 637-649.

[11] Joachims T. Text categorization with support vector machines: learning with many relevant features [C]// Proc of Machine Learning. 1998: 137-142.

[12] Zhou X, Tuck D P. MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data [J]. Bioinformatics, 2007, 23 (9): 1106-1114.

[13] Platt J. Sequential minimal optimization: a fast algorithm for training support vector machines [C]// Advances in Kernel Methods-Support Vector Learning. 1998.

[14] Hearst M A, Dumais S T, Osuna E, et al. Support vector machines [J]. IEEE Intelligent Systems and Their Applications, 1998, 13 (4): 18-28.

[15] Chen Y W, Lin C J. Combining SVMs with various feature selection strategies [J]. Studies in Fuzziness & Soft Computing, 2005, 207: 315-324.

[16] Powers D M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation [J]. Journal of Machine Learning Technologies, 2011, 2 (1): 37-63.

[17] XSSED [OL]. (2017-06) . http://xssed. com.

[18] XSS (cross site scripting) cheat sheet [EB/OL]. (2017-06) . http://ha. ckers. org/xssAttacks. xml.

[19] exploit-db [EB/OL]. (2017-06) . http://www. exploit-b. co/webapps.

[20] 甘俊英，张有为．一种基于奇异值特征的神经网络人脸识别新途径 [J]. 电子学报, 2004, 32 (1): 170-173.

[21] Han J, Pei J, Kamber M, et al. Data mining: concepts and techniques [M]. 2011.

[22] Witten I H, Frank E, Hall M A, et al. Data mining: practical machine learning tools and techniques [M]. 2016.

[23] 段宏湘，张秋余，张墨逸，等．基于归一化互信息的 FCBF 特征选择算法 [J]. 华中科技大学学报：自然科学版, 2017, 45 (1): 52-56.

[24] Kim S, D'Haro L F, Banchs R E, et al. The fourth dialog state tracking challenge [M]// Dialogues with Social Robots. 2017: 435-449.